AFRL-HE-WP-TR-2006-0048

AIR FORCE RESEARCH LABORATORY

Cognitive Systems Engineering Tool Survey – A Subtask in Support of Commander's Decision Aids for Predictive Battle-Space Awareness (CDA4PBA)

Mary Sanders
Elisabeth Fitzhugh

Adroit C4ISR Center
SRA International, Inc.
5100 Springfield Street, Suite 210
Dayton, OH 45431

January 2005

Final Report for February 2004 to December 2005

Approved for public release; distribution is unlimited.

Human Effectiveness Directorate
Warfighter Interface Division
Cognitive Systems Branch
2698 G Street
Wright-Patterson AFB OH 45433-7604

NOTICE

This report was cleared for public release by the Air Force Research Laboratory Wright Site (AFRL/WS) Public Affairs Office (PAO) and is releasable to the National Technical Information Service (NTIS). It will be available to the general public, including foreign nationals.

National Technical Information Service
5285 Port Royal Road, Springfield VA 22161

Federal Government agencies and their contractors registered with Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
8725 John J. Kingman Rd., STE 0944, Ft Belvoir VA 22060-6218

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2006-0048

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

FOR THE DIRECTOR

//SIGNED//

MARIS M. VIKMANIS
Chief, Warfighter Interface Division
Air Force Research Laboratory

This report is published in the interest of scientific and technical information exchange and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* January 2005	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* February 2004 - December 2005

4. TITLE AND SUBTITLE Cognitive Systems Engineering Tool Survey - A Subtask in Support of Commander's Decision Aids for Predictive Battle-Space Awareness (CDA4PBA)	**5a. CONTRACT NUMBER** FA8650-04-D-6405
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 63231F
6. AUTHOR(S) Mary Sanders, Elisabeth Fitzhugh	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER 2830HXC2
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Adroit C4ISR Center SRA International, Inc. 5100 Springfield Street, Suite 210 Dayton, OH 45431	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Materiel Command Air Force Research Laboratory Human Effectiveness Directorate Warfighter Interface Division Cognitive Systems Branch Wright-Patterson AFB OH 45433-7604	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFRL/HECS **11. SPONSOR/MONITOR'S REPORT NUMBER(S)** AFRL-HE-WP-TR-2006-0048

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distributed is unlimited.

13. SUPPLEMENTARY NOTES

Clearance AFRL/WS-06-1526, 15 Jun 06.

14. ABSTRACT

The objective for the cognitive systems engineering (CSE) tool survey task under Information Warfare Effectiveness (IWE) DO 6, Commander's Decision Aids for Predictive Battle-Space Awareness (CDA4PBA) is to identify system requirements necessary to capture the entire software and systems engineering process, from concept (including cognitive systems engineering) through product development, and to determine how well, through market research and in-house Subject Matter Experts (SMEs), existing commerical-off-the-shelf (COTS), government-off-the-shelf (GOTS), and research and development-based tools achieve those requirements.

15. SUBJECT TERMS Awareness, Battle-Space, Decision-Making, Military Operations, Predictive, Strategy and Tactics

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Janet E. Miller
a. REPORT UNCLASSIFIED	**b. ABSTRACT** UNCLASSIFIED	**c. THIS PAGE** UNCLASSIFIED	SAR	22	**19b. TELEPHONE NUMBER** *(include area code)*

THIS PAGE LEFT INTENTIONALLY BLANK

TABLE OF CONTENTS

TABLE OF FIGURES

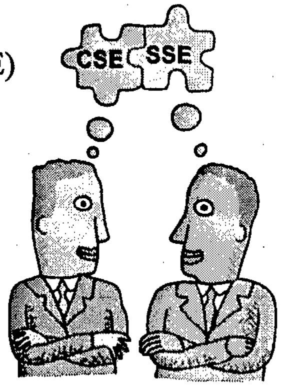
1 Introduction

The objective for the cognitive systems engineering (CSE) tool survey task under IWE DO 6 (CDA4PBA) is to identify system requirements necessary to capture the entire software and systems engineering process, from concept (including cognitive systems engineering) through product development, and to determine how well, through market research and in-house SMEs, existing COTS, GOTS, and R&D-based tools achieve those requirements.

1.1 Problem Statement

A software tool or process does not exist designed specifically to integrate CSE as a front-end technique to the software and systems engineering (SSE) process. As pointed out in the *Agile CSE* briefing (slide 21), a true working relationship must be developed between CSE and software development methods. The briefing noted three relationships:

- **Appeasement:** If you do "Y", we'll do "X"...
- **Synergy:** CSE works best with method Z. Let us show you how...
- **Revolution:** Software development should be conceived of and carried out in this fashion...



1.2 Constraints

Two main constraints were identified for this task:

Time/Schedule – with the limited time available, we were unable to obtain, set up, and use demo versions of the identified tools to allow us in-depth analysis of available tools. Our tool evalation is based on written documentation and interviews with in-house SMEs and tool customer service technicians.

Cost/Funding – as with any project, the time limitation for this project was driven by the amount of funding.

2 Approach

We took the following approach to look at the problem set, and then broke the work schedule into four phased work packages, as described below.

- What does the CSE methodology bring and require?
- What are the requirements for integrating CSE and SSE?
- What CSE and SSE tools exist today?
- How well do these current tools meet the requirements for an integrated tool?
- Understand CSE and SSE requirements
- Determine requirements for a CSE-integrated SSE framework
- Leverage SSE tool knowledge with in-house SMEs

- Identify shortfalls in current tools to the requirements
- Summarize

In Phase One, a representative set of steps describing the software and systems engineering (SSE) process yielded an SSE-oriented set of tool requirements. Concurrently, a working definition of cognitive systems engineering (CSE) and a representative set of CSE activities were compiled. The CSE effort differed from the SSE, in that the SSE process is defined for the complete software development lifecycle. In contrast, CSE may be involved from project initiation and continue, in partnership with SSE, through the full software development lifecycle, or it may contribute only through interface design and testing, depending on the project plan. Another CSE distinction is the lack of a single, definitive CSE process; practitioners are familiar with and may utilize any of a number of methods to define the work environment and human/system requirements. In consequence, the CSE activity list is deliberately inclusive of multiple CSE perspectives (Appendix I) in order to be of maximum utility. It is presented as a cafeteria-style offering from which a practitioner may choose as few or as many activities as seem appropriate to a given effort.

In Phase Two, a Scientific and Technical Information (STINFO) review identified representative sets of SSE and CSE tools. The Air Force Research Laboratory (AFRL) Technical Library and the AFRL XP/TT Independent Research and Development (IR&D) group located relevant government-sponsored academic and industry programs. An independent Internet search was conducted to identify existing and emerging commercial applications.

Phase Three evaluated the directed and independent search results; the team reviewed commercial literature and interviewed subject matter experts to determine which tools met requirements developed in Phase One and to identify gaps in tool capabilities.

Finally, in Phase Four, a subset of tools from Phase Three was assessed to determine which best met both SSE and CSE requirements.

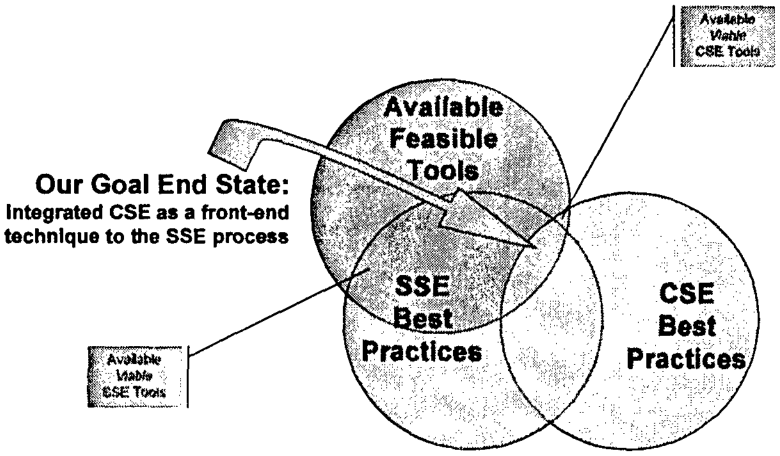

Available Viable CSE Tools
Available Feasible Tools
Our Goal End State:
Integrated CSE as a front-end technique to the SSE process
SSE Best Practices
CSE Best Practices
Available Viable SSE Tools

Figure 1. Our End Goal is to integrate CSE as a front-end technique to the SSE process.

The tool evaluation team worked to identify where CSE requirements might overlap with viable SSE tools (Figure 1), which means that a viable SSE tool must be extensible and expandable to absorb CSE methodology, and enable the user to automate as much front-end (CSE) work as possible.

2.1 Cognitive Systems Engineering (CSE)

2.1.1 CSE Activities Identification

CSE is an emerging discipline that models and evaluates the performance of interactive systems within a dual context of overall work accomplishment and specific task execution. CSE focuses on the cognitive aspects of the human/system interaction

Cognitive systems engineering (CSE) is both a field of scientific study and an approach to human-centered engineering for the design of interactive systems (Eggleston, 2002). CSE is an emerging discipline that models and evaluates the performance of interactive systems within a dual context of overall work accomplishment and specific task execution. In the context of software development, CSE's focus is the analysis of the system formed by the human-machine ensemble; CSE emphasizes the role of human cognitive processing within the *situational*, *system* and *environment* construct. This three-fold construct is specified through *work domain analysis*. CSE-derived insights lay the groundwork for software function and interface design requirements, identify use case elements, provide the basis for workload assessment, and supply the rationale for performance testing.

A range of CSE methods published in CSE texts and research reports were assessed to identify common CSE activities upon which to base CSE tool requirements. Two on-line resources that were of great value were the MITRE Survey of Cognitive Engineering Methods and Uses (http://mentalmodels.mitre.org/cog_eng/index.htm) and the Navy's SC-21/ONR Manning and Affordability Initiative's Human Engineering Resource (http://www.manningaffordability.com/S&tweb/Index_hse.htm). An integration of ideas from these sites with thoughts on Cognitive Systems Engineering (Rasmussen, Pjetersen, & Goodstein, 1994), Cognitive Work Analysis (Vicente, 1999), and the future of CSE and Work Centered Support System design (Eggleston, 2002) formed the basis for the high-level description of CSE support to software development in Figure 2. Please note: This set of activities is iterative and nonlinear—i.e., modeling and rapid prototyping are done repeatedly throughout the analysis, design and assessment phases as required.

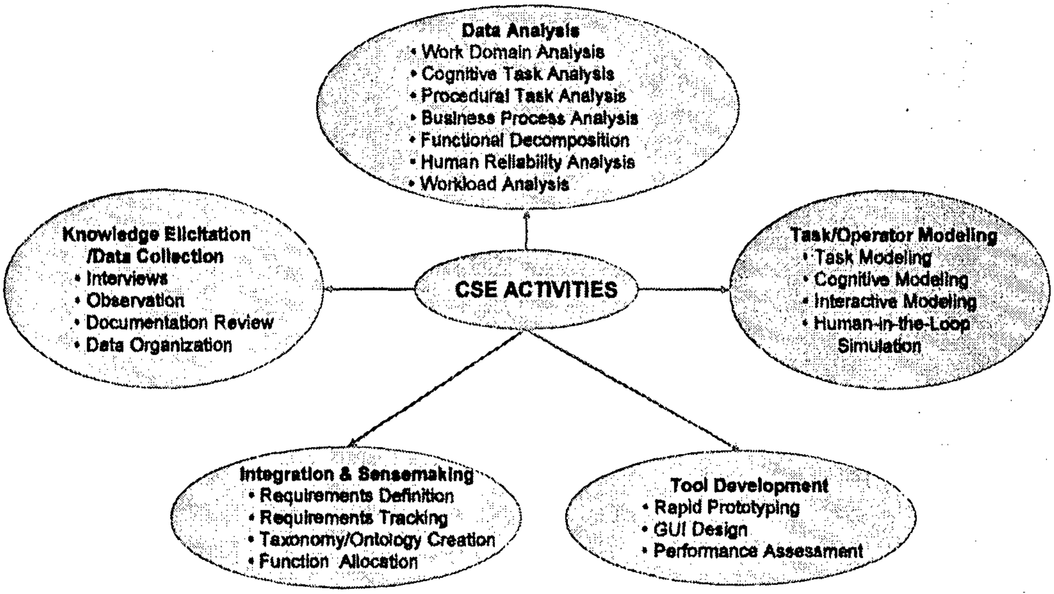


Figure 2. CSE Activities supporting software development.

The concept map in Figure 3 is from MITRE's Mental Models site; it illustrates the relationship between CSE areas of study—cognitive and behavioral processes, human error, and human-machine interaction—and CSE methods (http://mentalmodels.mitre.org/cog_eng/index.htm). Clearly, CSE's contribution to the software lifecycle falls primarily in development of requirements and testing to ensure those requirements are met; its apparent value lies in the breadth and rigor of its analytical process. However, CSE provides another critical function—system design is derived from analysis of the human-machine ensemble in a fully specified work context. CSE applies a thorough understanding of the worker and work domain to envision methods and allocate functions to enhance the business process, providing performance benefits beyond simple task automation.

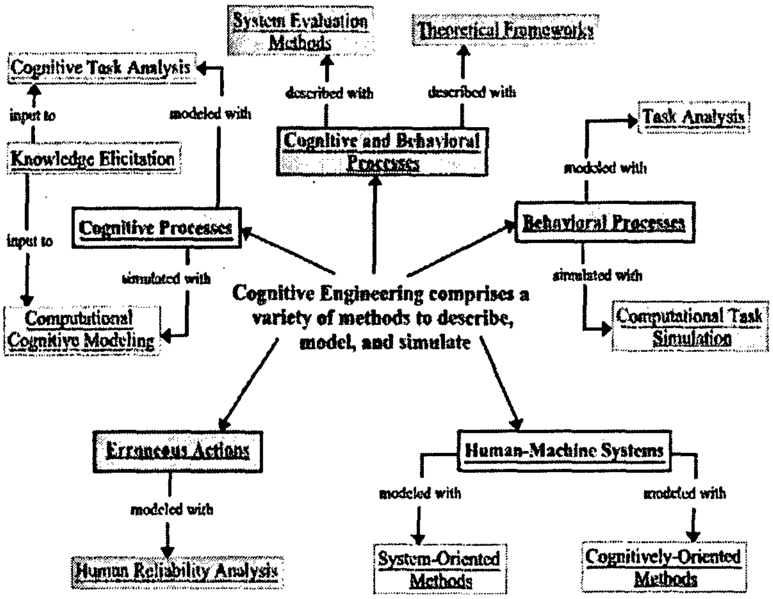


Figure 3. Cognitive Engineering Methods (MITRE).

2.1.2 CSE Tool Evaluation Methodology

The CSE tool evaluation used a detailed version of the CSE activities list and compared functions with support provided by tools found during the search conducted in Phase Two. This required some projection, as no required set of tool functionalities are associated with CSE activities (developing that list is beyond the scope of this effort). Tools were included in the evaluation based on whether the tool met at least one of two criteria:

1. Capabilities descriptions indicated the tool was designed specifically for CSE use
2. Tool literature described a function that fit within CSE methods

A list of seventy tools was compiled[1]. The tools were grouped into twelve categories: 1) Concept Mapping, 2) Observational Analysis, 3) Task Analysis, 4) Cognitive Modeling, 5) Human Performance Modeling, 6) Functional Decomposition Modeling, 7) UML Modeling, 8) Ontology Development, 9) GUI Development, 10) Usability Testing, 11) Requirements Management, and 12) Enterprise Architecture. Several tools were dropped from consideration immediately, either because they were government-sponsored research efforts that had lost funding and were not available (e.g., Work Domain Analysis Work Bench [WDAW], Computer Aided Cognitive Systems Engineering [CACSE]).or because they were not sufficiently developed (e.g., MacSHAPA). Other entries were limited to a few example candidates because their category was extremely large and significantly overlapped software development tools (e.g., GUI Design was limited to several representative high-end rapid prototyping tools). Finally, both IDEF and UML modeling tools and Requirements Management packages were dropped from final consideration because their capabilities were represented in the Enterprise Architecture tools. This limited the tool evaluation to some 50 candidates.

The evaluation procedure was straightforward. If the literature provided for a tool could be interpreted to mean that the tool supported each step in a CSE activity, the tool was rated fully capable. If the tool appeared to support some, but not all, of the steps, the tool was rated partially capable. If the tool did not appear to support an activity, it was not rated in that category.

The effort also included interviews with subject matter experts (SMEs). Four interviews were conducted, two with objected-oriented systems architects with both industrial and DoD software development backgrounds, and two with structured method/object-oriented system architects with extensive DoD software development and modeling and simulation expertise. The SMEs discussed their vision of the integration of CSE and software engineering and identified tools and tool capabilities that they believed provided value to both disciplines.

2.2 Software and Systems Engineering (SSE)

IEEE Standard Computer Dictionary, 610, ISBN 1-55937-079-3 (1990) defines software engineering as "The application of a systematic, disciplined, quantifiable approach to development, operation, and maintenance of software; that is, the application of engineering to software." Software engineering covers not only the technical aspects of building software systems, but also management issues, such as directing programming teams, scheduling, and budgeting.

[1] The MITRE CSE and the S&T Manning and Affordability CSE web sites both provided valuable assistance locating potential candidates for evaluation.

There are prescriptive software models, which define a distinct series of activities, actions, and tasks, as well as a workflow that can be used to build computer software. This may also be known as a system or software lifecycle. Some of the more recognizable models include Agile Process, Aspect-Oriented, Incremental, Concurrent, Rapid Prototyping, Rational Unified Process, Spiral, and Waterfall. For purposes of discussion, Figure 4 represents a generic Waterfall model.

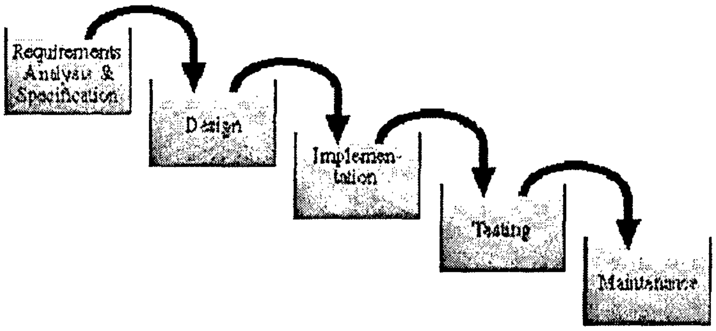


Figure 4. Waterfall Software Development Model.

Generally speaking, most software and system engineering models and methodologies begin with the basics – the requirements needed to build the system.

2.2.1 Transition from Analytical Insights to Design Requirements

There are as many methodologies and tools for software and systems engineering as there are companies who can perform these tasks. The crossover of *applicability* for interaction between CSE and SSE happens within the requirements engineering domain. Once the requirements have been explored, captured, documented, and shared, a software developer can proceed using whichever programming language, methodology, etc. is required.

2.2.2 Requirements Engineering—Integrating CSE and SSE

Generally speaking, requirements engineering is the process of establishing and documenting requirements – a collective set of one-line "shall" statements represent the things you need to know and the things you need to do that are critical for project success. A properly structured requirement must be a complete sentence that has one—and only one—interpretation. Each requirement should consist of a subject, verb, and predicate (no acronyms, no fragmentary statements). Requirements are the foundation for any development project and baseline documentation throughout the project life cycle. They serve as the starting point for traceability and provide the basis for test acceptance of the product. In its simplest form, requirements are written expressions of conditions, qualities, or capabilities to which the product must conform.

CSE provides value to the software development process through rigorous analysis of the human/system, its objectives, and its work environment. The rigor of the upfront analysis provides a sound framework both for requirements development and for function allocation. Comprehensive requirements and understanding of the work environment also illuminate interface design. The analysis process is where CSE interfaces with software design. CSE experts collect and analyze information and develop a mental image of what the system should look like and a concept of how it should function; however, there is a major difficulty translating the breadth of the CSE's acquired insight into the concise, one-line declarative statements (e.g., requirements) passed to the SSE for software development.

The transition from analytical insights to design requirements is currently an art, not a science. The situation could be ameliorated through the creation of guidelines and templates to shape the transfer of information; such aids may not enhance analytical ability but would certainly improve transmission and translation of critically important information.

MITRE has already mapped where CSE fits into software engineering phases (http://mentalmodels.mitre.org/cog_eng/ce_sys_eng_phases_matrix.htm). We have included the requirements analysis and review phases in this document. Communication among software developers is eased by a common frame of reference provided by a standard set of diagrams and established protocols; CSE adoption of similar protocols would enhance intercommunication.

Requirements Analysis Phase Objectives	**Cognitive Engineering Activities**
Develop system requirements and specifications.	• Determine current system capabilities and deficiencies. • Develop human performance, usability, and learnability requirements. • Develop decision-making and information requirements.

Most Applicable Cognitive Engineering Methods

- ACTA and the Critical Decision Method can be used to uncover the information, cues, and strategies required to make key decisions.
- Goal-directed task analysis can be used to uncover the information required to maintain situation awareness.
- Hierarchical Task Analysis can be used to record system requirements and how they can be achieved, including the order in which tasks and subtasks must take place.
- Interviewing and observing techniques, such as Contextual Inquiry, can be used to determine how well existing systems support user requirements.
- The results of the work domain analysis phase of CWA and ACWA can be used to develop requirements, and to help put the system requirements in context by providing a framework for determining whether requirements relate to the purpose of the system or to the physical devices of the system. The work domain analysis phase can also help define hardware and software needs (e.g. models, databases, sensors, etc.). In addition, the ACWA functional abstraction hierarchy can show the key decisions that must be made and the information required to make those decisions.
- The ANALYSE phase of COADE can be used to specify the cognitive requirements that must be addressed in system design.

Requirements Review Phase Objectives	**Cognitive Engineering Activities**
Throughout the system development process, review the system design with respect to its requirements and operational need.	• Determine whether the system effectively supports decision making and information needs of workers. • Determine whether the system design meets the human performance, usability, and learnability requirements. • Evaluate feedback from users to determine whether changes to system requirements are necessary. • Evaluate whether excessive performance and workload demands on personnel necessitate changes to system requirements.

Most Applicable Cognitive Engineering Methods

- Cognitive Task Analysis techniques, especially ACTA and GDTA, are useful in assessing whether the information needed to maintain situation awareness and make decisions are provided.

- Interviewing and observing and process tracing techniques can be used throughout the design process to get feedback from users as to how well a proposed design meets their needs.
- Computational Cognitive Modeling and Computational Task Simulation Techniques can be used to construct simulations of surrogate users interacting with a proposed system design, thus providing performance and workload estimates of a current design.
- Heuristic Evaluation, Walk-throughs/Talk-throughs/Cognitive Walk-throughs, Usability Studies, and Interface Evaluation Surveys can be used throughout the design process to evaluate how easy a proposed design is to use and learn.
- The work domain analysis phase of CWA and ACWA can be used to evaluate alternative designs in terms of how well the technical solution supports the functional purposes of the work domain. And CWA also allows testing of whether the system under development supports the necessary control tasks, strategies, role allocation and coordination structures, and operators' cognitive abilities.
- The EVALUATE phase of COADE can be used to verify whether the proposed system design actually addresses the requirements specified in the ANALYSE phase.

2.2.3 SSE Tool Evaluation Methodology

Because there are a plethora of SSE tools available, we first established baseline list of *feasible* SSE tools (see attached documentation.) To evaluate the tools, we performed literature searches and interviews, to best narrow to baseline list of *viable* SSE tools. For the purpose of this task, *feasible* means reasonably suitable for the purposes of blending CSE and SSE (e.g., your basic SSE tool suite), and *viable* means capable of putting into use for the purposes of blending CSE and SSE (i.e., extensible and expandable.)

The evaluation team reviewed SSE tools based on ten main criteria:

- Support Quality System Design
- Support Multiple System Views
- Methodology Independent
- User Interface (GUIs)
- Communication with Other Tools
- Document Production
- Computer Environment
- Resource Requirements
- User Interfaces (Multi-Tasking)
- Support and Maintenance

Ultimately, identified viable SSE tools must be extensible and expandable to absorb CSE methodology, and enable the user to automate as much front-end (CSE) work as possible.

3 Results Matrix

The results for the best fit, Enterprise Architecture (EA) tools are displayed in Figure 5 below.

Attribute/Tool Name	SE & Enterprise Architecture	CORE	Corporate Modeler Enterprise Ed.	System Architect (DoDAF)	METIS (DoDAF)	Enterprise Architect for DoDAF	Pro*Vision* Modeling Suite	Software Development Platform
1.0 Knowledge Elicitation								
1.1 Elicitation Planning								
1.2 Collection		Part	Part	Part	Part	Part	Part	Part
2.0 Observational Data Collection								
2.1 Observation Planning								
2.2 Collection								
3.0 Data Analysis								
3.1 Analytical Methods		Part	Part	Part	Part	Part	Part	Part
3.2 Integration / Sensemaking		Part	Part	Part	Part	Part	Part	Part
3.3 Requirements Tracking		Full	Full	Full	Full	Full	Full	Full
4.0 Facilitates Task/Operator Modeling								
4.1 Cognitive Modeling		Part	Part	Part	Part	Part	Part	Part
4.2 Task Modeling		Full	Full	Full	Full	Full	Full	Full
4.3 Interactive Modeling								
5.0 Facilitates Tool Development & Evaluation								
5.1 Rapid Prototyping								
5.2 GUI Design								
5.3 Performance Assessment								
5.4 Behavioral Modeling								
Interface w/Doors		?	Yes	Yes	Yes	Yes	?	Own

Figure 5. Collapsed view of the evaluation results matrix, showing EA tool capabilities.

4 Discussion

Few CSE-driven tools were found. The majority of CSE-focused tools (concept mapping, flow charting, and task analysis aids), were single focus and limited in scope. Eleven tools supported concept mapping and/or mind mapping. Several programs provided flow charting capabilities; a single commercially available tool was devoted to hierarchical task analysis.

Based on the search results, it appears that the majority of serious tool development has been focused on modeling cognitive processes and creating human performance modeling tools. These efforts have a dual goal: to further cognitive science research and to provide a means to test complex systems (e.g., new weapons systems) prior to deployment. The other most developed area was performance modeling—these efforts appear to be designed to support DoD and NASA development needs. Many are pilot or crew modelers that model individual and crew actions in complex tasks; some include workload assessment capabilities. Their utility to software application design depends on the intended use of the software—for example, pilot/system interactions are well-supported.

However, many software development tools were found applicable to CSE analysis. Universal Machine Language (UML) modeling and Integrated Computer-Aided Manufacturing Definition (IDEF) tools provided a broad range of diagramming support. UML modeling diagrams include Case, Package, Sequence, Activity, Class/Object, and State Diagrams. IDEF methods include IDEF0 (Activity Diagram), IDEF1 (Information Flow), IDEF3 (Process Flow and Object State Description), IDEF4 (Object-oriented Design Method) and IDEF5 (Ontology Description). A number of separate ontology-development tools support CSE work domain specification. Ontologies model a particular field of knowledge—the concepts and their attributes, as well as the relationships between the concepts.[2] GUI design was supported by several powerful rapid prototype packages (*Altia Graphics*, Disti *GL Studio*, and Engenuity Tech's *VAPS Designer* and *Developer*), providing drag-and-drop or menu-based GUI design, animated simulation and code export capabilities. Only two usability tools were identified. One, *Morae* (Techsmith), provides complete human/system interaction capture (video and keystroke) and analysis. The other, *Criterium Decision Plus* (InfoHarvest), applies an algorithm to ranked and weighted decision criteria to compare performance.

The most versatile and powerful capabilities were found in Enterprise Architecture (EA) tool suites. These tools combine both UML and IDEF software modeling with standardized views (templates) that provide a common frame of reference. They are typically fully integrated with requirements management and software development tools to provide an integrated development environment for all participants. The EA tool suites are also extensible. Open architecture and application programming interfaces (APIs) permit users to customize existing templates or create new ones.

In SME interviews, expert opinion converged on EA tool suites as providing the maximum flexibility within a single tool suite. All agreed that most task analysis/process analysis diagramming functions could be accomplished either within existing templates or with minor adjustments. The SMEs envisioned CSE to provide leadership in requirements definition, functional allocation, interface design and performance testing. EA tools were anticipated to provide an integrated development environment fostering collaboration between CSE analysts and software developers.

The successful EA tools were Department of Defense Architecture Framework (DoDAF) capable, providing all DoDAF modeling templates as add-ins. The single most capable tool was judged to be *Metis* (Computas); *Metis* integrates with other high-end software development tools (e.g., Telelogic's *DOORS* requirements management package). *Metis* was judged most flexible because it provides a wealth of templates and also features an internal GUI for template modification and creation.

5 Gaps

During the process of investigation for this tool evaluation, we discovered that tools that support collection planning and the actual collection itself are missing. While it would be most advantageous to employ a CSE expert for every software development effort, it would be most effective to have a tool that guides us through the collection process.

In all software development processes, in particular those that are CSE-driven, a wealth of information is gathered. The implementer generally hand-draws the flow charts or data to depict communications flows, constraints, data sinks, etc. The tools that are used during this process do not assist, guide, or augment the user's understanding of the material.

A lot of environmental information is also collected during the CSE process, and there is not currently an application that depicts work-arounds and alternatives, to include artifacts developed by the user, to assist them in getting the job done successfully.

[2] In software development, an ontology is represented as a set of classes with their associated slots. According to Gartner Research, "The objective of an ontology is to provide a formal specification of part of the real world..."

Finally, and potentially most importantly, there is a gap in the communication of the requirements collected by the CSE. CSE processes do not tend to depict requirements in the same manner a software developer needs to move forward with the actual programming of the software package.

6 Conclusions

A number of tools have been created that support CSE processes, but typically those tools are narrow in scope, have limited availability or support, and lack extensibility. Therefore, many practitioners are left to create their own aids, adapting existing, common software tools. CSE still needs a common, if not standardized, set of tools up front to aid in collection and analysis. Further, practitioners require the ability to take CSE results and quickly develop concepts through rapid prototyping tools which are "user-friendly" to non-programmers. Easily created rapid prototypes would enhance the requirements definition process and improve tool usability.

The survey identified the following shortfalls in CSE tool availability:

- Insufficient support for data collection (concept mapping tools only)
- Insufficient support for cognitive and functional task analysis (only a single hierarchical task analysis tool was located)
- No tailored support for the larger scope of work domain analysis
- Lack of integrated tools across CSE activities
- Limited support for human performance modeling
    - Several high end human performance modeling/assessment tools for aerospace acquisition efforts exist but there is limited support for [battle] planning and [battle] management development efforts
- Lack of structured decomposition tools for decision making processes
    - Classic structured decomposition support focuses solely on activities and information flow but provides no integrated, tailored support for decision making and related cognitive effort
    - Cognitive modeling tools have no GUI (programming intensive) and are stand-alone products
- *C-SETA*, the cognitive engineering decomposition tool supporting functional abstraction network (FAN) analysis is limited to FAN creation; a flexible tool that provides support to multiple CSE analytic methods (e.g., abstraction hierarchy, FAN)—*and* integrates with EA tool suite (including requirements management packages) is needed
- Lack of integrated, automated tools to support rapid translation of CSE analysis and rapid transfer to requirements management systems
    - Flexible multi-purpose flowcharting and concept mapping capabilities should be integrated into EA tools to support the IDE concept; should integrate with requirements management package
    - Diagram templates specific to CSE analyses should be integrated into EA tools

- Insufficient usability testing tools (most efforts support web site usability; only one keystroke performance capture tool was identified)
- Lack of aiding systems with embedded knowledge to guide design; lack of algorithm-driven usability tools to identify design errors

The survey also found a lack of consensus on exactly what might be considered a flexible but definitive CSE process. The human engineering process provided by the SC 21/ONR S&T Manning and Affordability web site, shown in Figure 6 below, is directly analogous to the SSE process. However, it is less detailed than the Cognitive Engineering methods in Figure 3, which more closely resembles this efforts' vision of CSE. It is not cognitively focused, shows no process redesign orientation, and fails to specify workload and human reliability analysis, both of which are critical to usability.

The lack of consensus is expected to have a negative impact on conceiving and designing tools to support CSE. As the survey showed, few attempts have been made to support the initial knowledge elicitation/data collection activities and the multiple flavors of task analysis, while no support exists for identification of requirements for contextual help—a critical usability issue. Only one source—Cutter International—was found that provided a full range of support to workload and human reliability analysis, but the *ShipSHAPE* tool suite was not considered herein because it has not yet been fully migrated from Mac System 7 OS to Windows operating systems. However, their tools, developed under Navy and DARPA contracts, show great promise and if they continue to be updated, should be considered in any future evaluation effort.

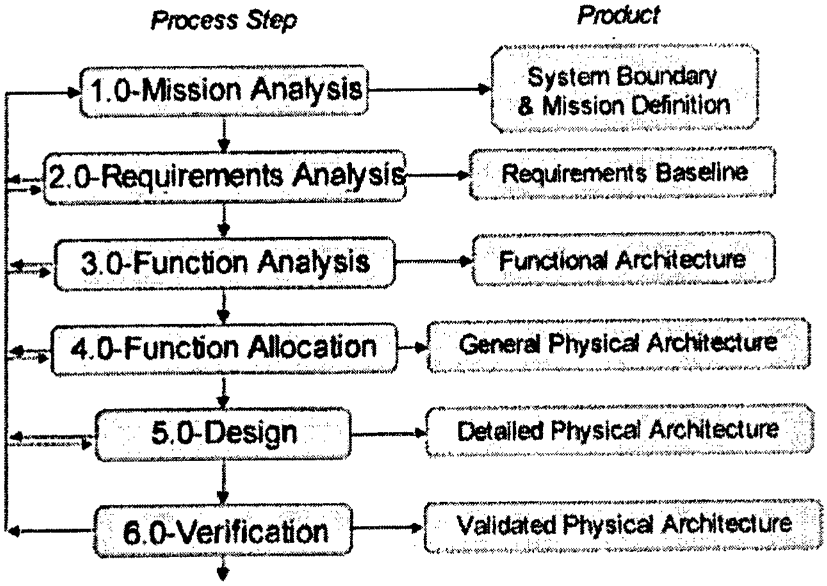


Figure 6. Human Engineering Process (SC-21/ONR S& T Manning and Affordability).

7 Recommendations

CSE provides value to the software development process through rigorous analysis of the human/system, its objectives, and its work environment. The rigor of the upfront analysis provides a sound framework both for requirements development and for function allocation. Comprehensive requirements and understanding of the work environment also illuminate interface design. The analysis process is where CSE interfaces with software design. The transition from analytical insights to design requirements is currently an art, not a science.

To enhance the transfer of analytical products, both CSE and SSE need to work in an Integrated Development Environment (IDE). EA tools bear the most promise; it is sensible and practical to integrate CSE processes into the tool set that is already in use by software developers. Of the tools on the market, *Metis* by Computas appears to offer the widest range of capability combined with the greatest flexibility. While several tools offer an API for extensibility, *Metis* alone provides a user interface within the tool. The evaluation team strongly recommends further evaluation of *Metis* to validate this conclusion and to determine what is required to customize and extend its existing capabilities to serve CSE.

7.1 Future Efforts

7.1.1 Refine CSE Process

The tool evaluation team perceives a need to refine the CSE process and collect information on commonalities in practitioner methods. This effort should also include a translation of process components to tool requirements. Clearly the resources were not available to conduct a needs analysis for tool preferences through practitioner interviews; however, this is a key element of any requirements definition and analysis. The analysis conducted herein was limited, in that, some 200 SSE and CSE tools were identified, but only a limited market analysis could be conducted. A more in-depth analysis of the best candidates from this survey would help focus efforts.

7.1.2 Further Investigation of Specific Tools

The tool evaluation team recommends further investigation of a number of COTS products that may be able to support the CSE-SSE integration.

CORE products – the *CORE* line of products (http://www.vitechcorp.com/overview.html) includes behavioral modeling constructs in addition to a full line of SSE tools.

Ship-SHAPE – The *Ship-SHAPE* Automated Human Systems Integration (HSI) tool suite by Carlow International (http://carlow.com/hsitools.html)was developed under contract with the US Navy and DARPA. The tool set includes HSI process, planning and assessment support and functional allocation, task analysis, and simulation tools. Most of the tool set is compatible with both Mac and Windows operating systems; an HTML version is in development.

Microanalysis & Design (http://www.maad.com/index.pl/products) offers a line of human performance modeling and simulation tools (e.g., *WinCrew*, *IPME*, *CSDT*) that incorporate individual and crew interaction modeling, workload modeling, and crew station design.

Morae – TechSmith (http://www.techsmith.com/products/morae) offers an all-digital, software-based product that records and synchronizes user and system data for software and web site usability analysis.

TaskArchitect (http://taskarchitect.com/) is designed to support hierarchical task analysis. It was not within the scope of the current task to determine whether it was flexible enough for use with other forms of task analysis.

7.1.3 Proof of Concept

SRA's tool evaluation team strongly recommends a proof-of-concept task which would enable us to perform further analysis and allow us to determine whether COTS tools would work together well enough to support a CSE front-end and an SSE back-end. At this time, the tool evaluation team recommends looking into a variety of potential paths, including using Word to develop CSE templates and hooking them directly into an SSE tool, and alternatively, using Word to develop CSE templates and hooking them into a requirements tool, then into an SSE tool. Potential pathways are depicted in Figure 7. We recommend connecting from a template developer (e.g., Word) directly to identified SSE tools, and also following pathways from a template developer through identified requirements management tools, then into identified SSE tools.

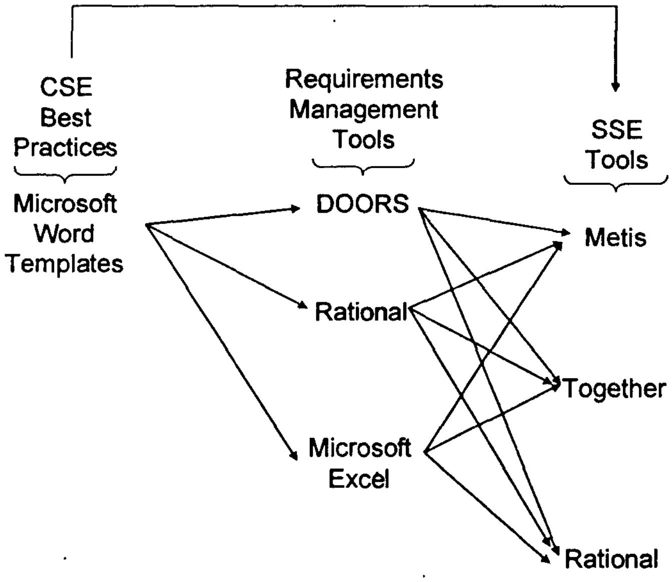


Figure 7. Potential pathways for proof-of-concept.

Appendix I. CSE Perspective

Cognitive systems engineering (CSE) is a field of scientific study and an approach to human-centered engineering for the design of interactive systems (Eggleston, 2002). An emerging discipline, CSE models and evaluates the performance of interactive systems within a dual context of overall work accomplishment and specific task execution. Interactive systems encompass interactions among people, information systems, and the organizations of which they are part; CSE examines the relationship among users, tools, tasks and the user's work domain (Vicente, 1999). CSE's central theme is that within the interactive system, the user-tool complex or *human-machine ensemble*, needs to be conceived, designed, analyzed, and evaluated as a *cognitive system.* A cognitive system is defined as a system that is able to completely or partially control its own behavior using prior or situation-specific information about itself. CSE emphasizes the role of human cognitive processing within the situational, system and environment construct. Cognitive processing and decision making are critical design issues in complex interactive systems.

Primary CSE research focuses are cognitive processes involved in decision-making and activity coordination; contributory and inhibitory effects of automated information systems (including decision-aiding tools); and the full range of human-computer interface design issues. CSE research techniques determine user requirements, model information flow (including information vulnerabilities and dependencies), and identify effective decision strategies and candidate areas for automated decision making support. Additionally, CSE provides a framework for investigation of the effects of confidence in automated decision aids on operator performance. It also researches and applies knowledge of human and machine strengths and vulnerabilities in order to design error resistant systems.

CSE methods focus predominately in one or more of three areas: the task, the user or the domain (Eggleston, 2002). CSE goals include the development of improved information systems and the effective integration of tools and data to enhance human task performance. CSE encompasses modeling methodologies from engineering, psychology, cognitive science, information science, management science, and computer science to develop concepts, methods, and tools for analyzing and designing useful and acceptable work systems based on orderly analysis of human cognitive tasks (e.g., perceptive and adaptive thinking for planning, organizing, collaborating, and problem-solving) (Roth, Patterson, & Mumaw, 2001). CSE design is grounded in early and accurate assessment of both implicit and explicit user requirements. The application of cognitive systems engineering enables system developers to accurately (and comprehensively) identify and integrate all system requirements early in the design process, saving time, effort, and expense.

Cognitive systems engineering can be viewed as a blend of cognitive work analysis (CWA), cognitive task analysis (CTA), functional task analysis, human/system reliability analysis, and workload analysis. CSE also incorporates evaluation. Although the steps are discussed in a linear fashion, the whole process is actually iterative. Evaluation occurs several times throughout the development process as knowledge is accumulated and the project situation changes (Eggleston, 2002).

CWA, developed primarily by Rasmussen (1994), displays an integrated approach to human-centered system design. The approach is divided into five stages: 1) Work Domain Analysis, 2) Control Task Analysis, 3) Strategies Analysis, 4) Social Organization and Cooperation Analysis,

Appendix I. CSE Perspective

Cognitive systems engineering (CSE) is a field of scientific study and an approach to human-centered engineering for the design of interactive systems (Eggleston, 2002). An emerging discipline, CSE models and evaluates the performance of interactive systems within a dual context of overall work accomplishment and specific task execution. Interactive systems encompass interactions among people, information systems, and the organizations of which they are part; CSE examines the relationship among users, tools, tasks and the user's work domain (Vicente, 1999). CSE's central theme is that within the interactive system, the user-tool complex or *human-machine ensemble*, needs to be conceived, designed, analyzed, and evaluated as a *cognitive system.* A cognitive system is defined as a system that is able to completely or partially control its own behavior using prior or situation-specific information about itself. CSE emphasizes the role of human cognitive processing within the situational, system and environment construct. Cognitive processing and decision making are critical design issues in complex interactive systems.

Primary CSE research focuses are cognitive processes involved in decision-making and activity coordination; contributory and inhibitory effects of automated information systems (including decision-aiding tools); and the full range of human-computer interface design issues. CSE research techniques determine user requirements, model information flow (including information vulnerabilities and dependencies), and identify effective decision strategies and candidate areas for automated decision making support. Additionally, CSE provides a framework for investigation of the effects of confidence in automated decision aids on operator performance. It also researches and applies knowledge of human and machine strengths and vulnerabilities in order to design error resistant systems.

CSE methods focus predominately in one or more of three areas: the task, the user or the domain (Eggleston, 2002). CSE goals include the development of improved information systems and the effective integration of tools and data to enhance human task performance. CSE encompasses modeling methodologies from engineering, psychology, cognitive science, information science, management science, and computer science to develop concepts, methods, and tools for analyzing and designing useful and acceptable work systems based on orderly analysis of human cognitive tasks (e.g., perceptive and adaptive thinking for planning, organizing, collaborating, and problem-solving) (Roth, Patterson, & Mumaw, 2001). CSE design is grounded in early and accurate assessment of both implicit and explicit user requirements. The application of cognitive systems engineering enables system developers to accurately (and comprehensively) identify and integrate all system requirements early in the design process, saving time, effort, and expense.

Cognitive systems engineering can be viewed as a blend of cognitive work analysis (CWA), cognitive task analysis (CTA), functional task analysis, human/system reliability analysis, and workload analysis. CSE also incorporates evaluation. Although the steps are discussed in a linear fashion, the whole process is actually iterative. Evaluation occurs several times throughout the development process as knowledge is accumulated and the project situation changes (Eggleston, 2002).

CWA, developed primarily by Rasmussen (1994), displays an integrated approach to human-centered system design. The approach is divided into five stages: 1) Work Domain Analysis, 2) Control Task Analysis, 3) Strategies Analysis, 4) Social Organization and Cooperation Analysis, Easter, 2002). Cognitive systems engineering, in its simplest form, integrates human-systems engineering with the cognitive aspects of work to account for factors which significantly influence design for the operation of today's technologically complex systems. CSE as a science has grown from intangible user attributes to include cognitive processing activities. The process was then extended to the interface between the user and the system controlling mechanism and then to the user-system as a whole. Finally, today CSE has grown to include the system as well as extraneous factors influencing that system such as the environment, collaboration, and social and organizational factors.

Appendix II. References

Dominguez, Cindy et.al. (Klein Associates). (2004) *Agile CSE.*
International Journal of Man-Machine Studies, *18*, 583-600.

Eggleston, R. G. (2002). Cognitive Systems Engineering at 20-something: Where do we stand? 15-78. In M. D. McNeese & M. A. Vidulich (Eds.), *Cognitive systems engineering in military aviation environments: Avoiding Cogminutia Fragmentosa* (pp. 15-77). Wright-Patterson Air Force Base, OH: HSIAC Press.

Eggleston, R. (2003). Work-centered design: A cognitive engineering approach to system design. *Proceedings of the 47th Annual Meeting of the Human Factors and Ergonomics Society.*

Flach, J. M., & Kuperman, G. G. (1998). *Victory by design: War, information, and cognitive systems engineering,* (Tech. Rep. AFRL-HE-WP-TR-1998-0074). Wright-Patterson AFB, OH: U.S. Air Force Research Laboratory.

Hollnagel, E., & Woods, D. D. (1983). Cognitive systems engineering: New wine in new bottles,

IEEE Computer Society. (1997). *Software Requirements Engineering, Second Edition.* IEEE Computer Society Press, Los Alamitos, CA.

International Council on Systems Engineering (INCOSE), http://www.incose.org/.

MITRE. (no date). *A survey of cognitive engineering methods and uses.* Accessed on-line 24 January 2005 at: http://mentalmodels.mitre.org/cog_eng/index.htm.

Potter, S. S., Elm, W. C., Roth, E. M., Gualtieri, J. W., & Easter, J. R. (2002). Using intermediate design artifacts to bridge the gap between cognitive analysis and cognitive engineering. In M. D. McNeese & M. A. Vidulich (Eds.), *Cognitive systems engineering in military aviation environments: Avoiding Cogminutia Fragmentosa* (pp. 137-166). Wright-Patterson Air Force Base, OH: HSIAC Press.

Potter, S. S., Roth, E. M., Woods, D. D., & Elm, W. C. (1998). *Toward the development of a computer-aided cognitive engineering tool to facilitate the development of advanced decision support systems for information warfare domains* (Tech. Rep. AFRL-HE-WP-TR-1998-0004). Wright Patterson AFB, OH: Air Force Research Laboratory.

Rasmussen J., Pejtersen, A. M., & Goodstein, L. P. (1994). *Cognitive systems engineering.* New York: Wiley.

Roth, Gualtieri, Easter, Potter, & Elm, in preparation.

Roth, E. Patterson, E., & Mumaw, R. (2001). Cognitive engineering: Issues in user-centered system design. In J. Marciniak (Ed.), *Encyclopedia of software engineering* (2nd ed.). NY: John Wiley. Accessed on-line at: http://csel.eng.ohio-state.edu/emily/cog_eng_def.pdf.

SC21/ONR S&T Manning & Affordability (2001). *The Human Engineering Tools List.* Accessed on-line 24 January 2005 at: http://www.manningaffordability.com/S&tweb/HEResource/Tool/Tool_list.htm.

Vicente, K. (1999). *Cognitive work analysis: Toward safe, productive, and healthy computer-based work.* Mahwah, NJ: Lawrence Erlbaum.